

## PENGENDALIAN LCD GRAFIK 128\*64 TITIK BERBASIS MIKROKONTROLER

**Deddy Susilo**

Fakultas Teknik Program Studi Teknik Elektro dan Komputer  
Universitas Kristen Satya Wacana Salatiga  
Email : [sp\\_tech\\_eng@yahoo.co.id](mailto:sp_tech_eng@yahoo.co.id)

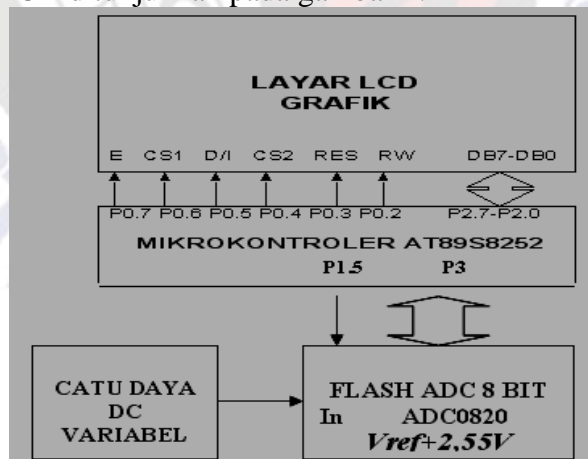
### Abstrak

*Liquid Crystal Display* atau LCD adalah salah satu revolusi dibidang elektronika optik yang berfungsi sebagai alat penampil. Makalah ini akan membahas contoh-contoh pengendalian dan pemrograman LCD grafik 128\*64 titik. Penjelasan akan dimulai dengan sisi rancang bangun pada perangkat keras (*hardware*) dan algoritma perangkat lunaknya (*software*).

**Kata kunci :** *Graphics LCD 128\*64 pixel, Crystalfonts Graphics LCD.*

### 1. Pendahuluan

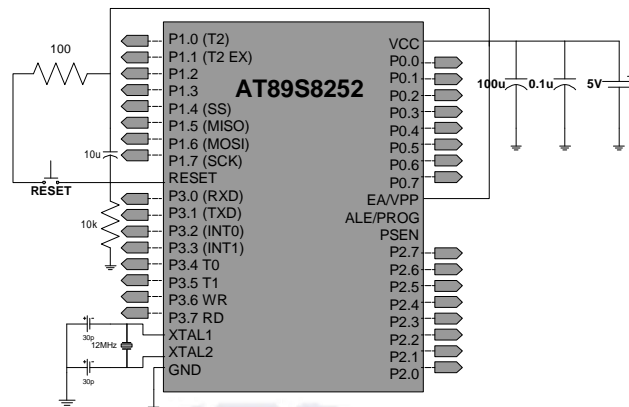
Berkreasi dengan LCD grafik dengan ukuran 128 \* 64 titik diberi contoh aplikasi sebagai penampil dari hasil pengukuran level tegangan DC dikendalikan dengan mikrokontroler. Prinsip dasar dari menampilkan di layar LCD adalah dengan mengakses titik-titik pada layar sesuai alamat memorinya. Bagan pengendalian mikrokontroler ke LCD ditunjukkan pada gambar 1.



Gambar 1. Hubungan Antara Mikrokontroler Dengan Modul LCD Grafik

LCD Grafik yang mempunyai lebar 128 titik akan digunakan untuk menunjukkan level tegangan masukan ADC dengan tegangan acuan pada ADC sebesar 2,55 voltDC. ADC yang digunakan mempunyai lebar data 8 bit sehingga keluaran maksimum adalah FF heksadesimal atau 255 desimal. Dengan demikian, tiap perubahan 10 milivoltDC pada masukan akan mengakibatkan kenaikan 1 bit pada keluaran ADC.

Mikrokontroler yang dipakai memiliki 8 Kbyte *Flash* ROM dimana program sistem pengendali modul LCD Grafik, pengambilan data hasil konversi analog ke digital oleh ADC 8 bit dan pengolahannya untuk dapat ditampilkan pada layar. Untai mikrokontroler ditunjukkan pada gambar 2.

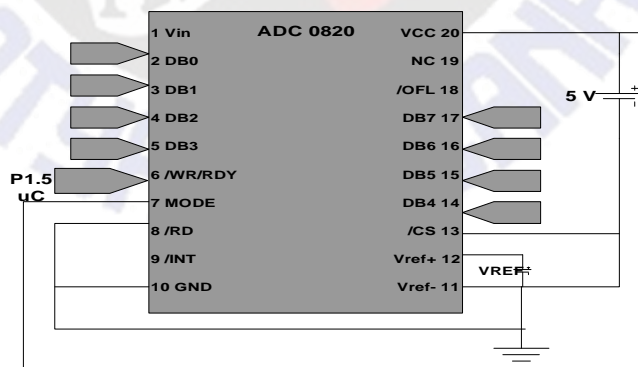


Gambar 2. Untai Sistem Minimum Mikrokontroler AT89S8252

Mikrokontroler akhir ini menggunakan pembangkit sinyal berupa osilator kristal sebesar 12 MHz sehingga memiliki kecepatan 1  $\mu$ sekon per siklus mesin. Jika instruksinya membutuhkan waktu 1 siklus mesin maka dalam 1 detik akan dapat mengeksekusi 1 juta instruksi. Instruksi-instruksi dalam MCS-51 membutuhkan waktu yang bervariasi dari 1  $\mu$ s hingga 3  $\mu$ s.

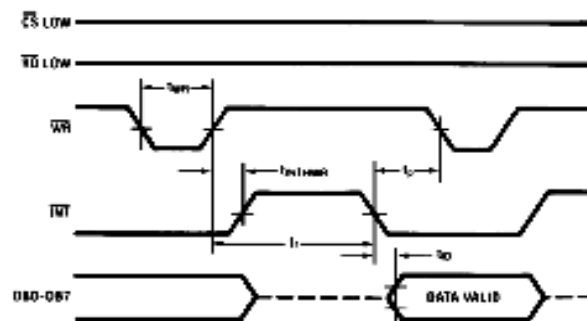
## 2. Untai Flash ADC 8 Bit

Untai ini berfungsi mengubah data analog dari masukan tegangan DC ke data digital untuk kemudian diolah di mikrokontroler. Untai ADC 0820 ini difungsikan dalam operasi *stand alone* yaitu dapat mengkonversi secara otomatis tanpa pengendalian dari luar. Mikrokontroler hanya mengendalikan 1 buah pin yaitu /WR atau *write* untuk memerintahkan ADC mengeluarkan data terbaru pada jalur data DB0 hingga DB7. Untai ADC ditunjukkan pada gambar 3.



Gambar 3. Untai Flash ADC 0820

Algoritma pengendalian ADC dijelaskan pada penjelasan diagram waktu ADC seperti pada gambar 4.

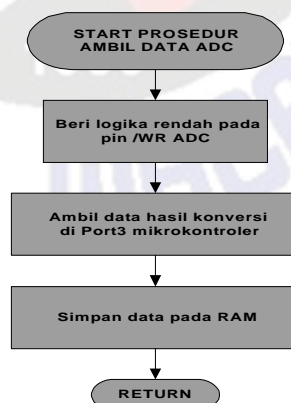


Gambar 4. Diagram Waktu ADC Pada Operasi Stand Alone

Untai ADC dikendalikan sesuai dengan diagram waktu pada gambar 4. Pin /CS atau *chip select* adalah pin pengendalian untuk mengaktifkan IC ini, jadi jika pin /CS diberi rendah (*low*) IC ADC tak akan bisa diakses untuk diambil datanya. Jalur data akan bernilai ambang (*high impedance*). Pin /RD diberi rendah (*low*) supaya ADC siap dalam kondisi dibaca datanya. Pin /WR adalah pin kendali dari mikrokontroler untuk memerintahkan ADC menulis atau mengeluarkan data. Data yang terambil saat pin /WR dibuat rendah pertama kali adalah data sebelumnya, sehingga untuk mendapatkan data baru pin /WR harus di buat rendah kembali setelah menunggu pin /INT bernilai tinggi (*high*) sebagai tanda selesai konversi. Kemudian data siap diambil dan disimpan dalam RAM mikrokontroler.

Algoritma perangkat lunak pengambilan data digital dari ADC adalah sebagai berikut dan diagram alirnya ditunjukkan pada gambar 5.

1. Pin P1.5 mikrokontroler dihubungkan pada pin /WR ADC.
2. Beri nilai *low* port P1.5 untuk mengendalikan pin /WR.
3. Ambil nilai pada jalur data 8 bit DB0-DB7 dengan port 3 mikrokontroler.
4. Simpan data dalam RAM mikrokontroler.
5. Kembali ke program utama.



Gambar 5. Diagram Alir Pengendalian dan Pengambilan Data ADC

### 3. LCD Grafik 128x64 titik

Salah satu bagian yang penting pada perancangan sistem ini adalah bagian yang berguna untuk menampilkan isyarat yang diukur. Untuk itu dipakai sebuah penampil modul LCD grafik 128 \* 64 *pixel*. Penampil modul LCD grafik ini memiliki 20 pin yang 14 diantaranya dihubungkan pada mikrokontroler, sedangkan 6 kaki lainnya dipakai sebagai tegangan catu positif, ground, mengatur *contrast* dari modul LCD dan mengatur lampu latar belakang (*backlight*).

Pembahasan modul LCD ini dibatasi pada bagaimana menampilkan isyarat yang diukur pada modul LCD grafik ini. Proses menampilkan meliputi proses *mapping* dan proses penulisan data pada modul LCD. Modul LCD yang dipakai terbagi menjadi dua bagian utama dengan ukuran 64 \* 64 titik dan terbagi lagi menjadi baris (dalam *byte*) dan kolom sesuai yang terlihat pada gambar 2.21. Jadi saat akan melakukan penulisan pada modul LCD harus ditentukan dahulu posisinya (*mapping*), baru kemudian dikirimkan data-data yang bisa menggambarkan bentuk isyarat yang diukur.

Tabel 1. Pin-pin pada Modul LCD Grafik

Nomor Pin	Simbol	Level	Kegunaan
1	VSS	0V	Tegangan catu nol volt
2	VDD	5V	Tegangan catu untuk LCD
3	V <sub>o</sub>	Variabel	Tegangan untuk kontras layar LCD
4	D/I	H/L	H : data L : instruksi
5	R/W	H/L	H : Baca (modul LCD ke mikrokontroler) ; L : Tulis (mikrokontroler ke modul LCD)
6	E	H	Sinyal enabel
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	CS1	L	Pemilih kolom 1—64
16	CS2	L	Pemilih kolom 65—128
17	RES	L	Sinyal reset
18	Vout		Tegangan negatif untuk kontras layar LCD
19	A		Tegangan catu positif untuk lampu latar belakang
20	K		Tegangan catu negatif untuk lampu latar belakang

Berikut ini dibahas instruksi yang dipakai untuk menulis pada modul LCD grafik. Secara garis besar ada dua macam instruksi, yaitu menuliskan perintah atau data.

### Menulis perintah atau data.

#### 1. Penampil *on* atau *off*.

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	D

Nilai data yang akan ditampilkan bisa tertampil pada layar penampil modul LCD grafik jika nilai D sama dengan satu (*high*) dan jika D bernilai 0 (*low*) maka tak ada data yang bisa tertampil pada layar modul LCD.

#### 2. Menentukan awal baris.

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	A	A	A	A	A	A

Berfungsi untuk menentukan awal dari baris yang bisa menampilkan data.

#### 3. Menentukan baris (*X address*).

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	1	1	A	A	A

Pengalamatan X AAA (biner) dari data RAM untuk layar penampil ditentukan melalui register pengalamatan X. Setelah itu, menulis atau membaca data dilakukan pada baris yang telah ditentukan ini sampai baris yang baru ditentukan.

#### 4. Menentukan kolom (Y address)

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A	A	A	A	A	A

Pengalamatan Y AAAAAA (biner) dari RAM data untuk layar penampil ditentukan melalui register pengalamatan Y. Setelah itu nilai register ini akan dinaikkan satu setiap dilakukan pembacaan nilai data atau menuliskan data. Proses menentukan pengalamatan X dan Y sering disebut juga dengan proses *mapping*.

#### 5. Menuliskan data penampil

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	D	D	D	D	D	D	D	D

Menuliskan 8 bit data DDDDDDDD pada RAM data untuk penampil. Setelah itu nilai Y akan naik satu secara otomatis.

#### Menulis data atau membaca status

##### 1. Membaca data penampil

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D	D	D	D	D	D	D	D

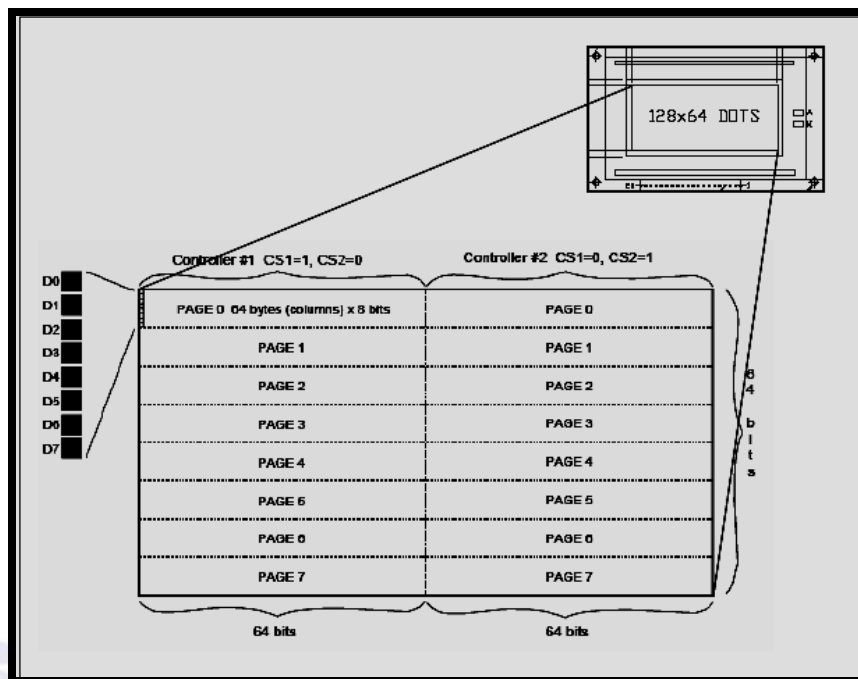
Membaca data 8 bit DDDDDDDD (biner) dari RAM data untuk penampil. Lalu nilai alamat Y akan dinaikkan 1 secara otomatis.

##### 2. Membaca status

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	<i>Busy</i>	0	<i>On / Off</i>	<i>RESE T</i>	0	0	0	0

- **Busy**  
Saat bit *busy* bernilai satu maka LSI sedang mengeksekusi instruksi terakhir yang diberikan dan belum bisa menerima atau menjalankan instruksi yang baru. Jadi sebelum memberikan instruksi selanjutnya kita harus yakin bahwa bit *busy* bernilai nol.
- **ON / OFF**  
Bit ini menunjukkan kondisi LCD dalam keadaan *on* atau *off*. Saat bit ini bernilai satu maka layar dalam keadaan *on*, dan jika bit ini bernilai nol maka menunjukkan bahwa layar dalam keadaan *off*.
- **Reset**  
Jika bit ini bernilai satu, hal ini menunjukkan sistem sedang diinisialisasi. Pada saat keadaan ini, tidak ada instruksi yang bisa diterima kecuali instruksi untuk membaca status. Jika bit ini bernilai nol maka menunjukkan bahwa proses inisialisasi telah selesai dilakukan dan siap menerima instruksi selanjutnya.





Gambar 6. Pemetaan Layar LCD Grafik 128\*64 titik

Secara garis besar layar LCD yang berukuran 128x64 titik ini dibagi menjadi 2 layar seperti yang ditunjukkan pada gambar 6, untuk mengakses layar sebelah kiri atau layar1 pin CS1 harus diberi nilai rendah dan CS2 diberi nilai tinggi, demikian sebaliknya jika ingin mengakses layar2. Untuk masing-masing layar dibagi lagi menjadi 8 buah halaman yaitu PAGE0 hingga PAGE7 yang diberi simbol X=0 hingga X=7. Untuk kolom sebanyak 64 per layar diberi nama Y=0 hingga Y=64. Mengakses LCD ini harus dilakukan per halaman X dan per kolom Y, jadi sekali mengakses harus mengakses 8 titik pada X tertentu dan Y tertentu. Untuk mengakses 2 layar sekaligus tidak bisa dilakukan dalam satu waktu melainkan bergantian tiap layar. Berikut subrutin-subrutin dasar dalam LCD grafik.

**Subrutin deklarasi global variable:**

```
DATA_LCD EQU P2 ; DB7-DB0 pada modul LCD à Port 2
ENABLE BIT P0.7 ; Pin ENABLE modul LCD à Port 0.7
DI BIT P0.5 ; Pin DI dengan Port 0.5
CS1 BIT P0.6 ; Chip Select Layar 1 untuk kolom 0-63
CS2 BIT P0.4 ; Pin Chip Select Layar 2 untuk kolom 64-127
RES BIT P0.3 ; Pin RESET dengan Port 0.3
RW BIT P0.2 ; Pin Read/Write dengan Port 0.2
```

**Subrutin memilih layar:**

```
//memilih layar 1
LCD1:
    CLR CS2
    SETB CS1 ; CS2 = '0' dan CS1 = '1'
    CLR DI ; DI = '0' à instruksi
    CLR RW ; RW = '0' à tulis
    SETB ENABLE ; ENABLE = '1'
    MOV DATA_LCD,#3FH ; perintah menghidupkan layar
    CLR ENABLE ; ENABLE = '0'
    CALL CEK_LCD ; lompat ke subrutin cek_lcd, apakah sibuk?
    RET ; return

//memilih layar 2
LCD2:
```

```
CLR CS1
SETB CS2          ; CS2 = '1' dan CS1 = '0'
CLR DI            ; DI = '0' à instruksi
CLR RW           ; RW = '0' à tulis
SETB ENABLE      ; ENABLE = '1'
MOV DATA_LCD,#3FH ; perintah menghidupkan layar
CLR ENABLE       ; ENABLE = '1'
CALL CEK_LCD     ; lompat ke subrutin cek_lcd, apakah sibuk?
RET              ; return
```

**Subrutin memilih kolom pertama dari layar:**

```
SET_COLUMN0:      ; cara menempatkan posisi ke kolom 0
CLR DI            ; DI = '0' à instruksi
CLR RW           ; RW = '0' à tulis
MOV DATA_LCD,#01000000B
SETB ENABLE      ; ENABLE = '1'
CLR ENABLE       ; ENABLE = '0'
CALL CEK_LCD     ; lompat ke subrutin cek_lcd, apakah sibuk?
RET              ; return
```

**Subrutin memilih halaman pertama dari layar:**

```
SET_PAGE0:        ; cara menempatkan posisi ke halaman 0
CLR DI            ; DI = '0' à instruksi
CLR RW           ; RW = '0' à tulis
MOV DATA_LCD,#10111000B
SETB ENABLE      ; ENABLE = '1'
CLR ENABLE       ; ENABLE = '0'
CALL CEK_LCD     ; lompat ke subrutin cek_lcd, apakah sibuk?
RET              ; return
```

**Subrutin menulis 1 byte atau 8 piksel pada layar LCD:**

TULIS:

```
SETB DI           ; DI = '1' à data
CLR RW           ; RW = '0' à tulis
SETB ENABLE      ; ENABLE = '1'
CLR ENABLE       ; ENABLE = '0'
RET              ; return
```

**Subrutin pengecekan status kesibukan modul LCD:**

```
CEK_LCD:          ; cek apakah LCD dapat menerima instruksi berikutnya
CLR ENABLE       ; ENABLE = '0'
CLR DI           ; DI = '0' à instruksi
SETB RW         ; RW = '1' à baca
SETB ENABLE     ; ENABLE = '1'
JB P2.7,CEK_LCD ; jika Port 2.7 masih '1' looping ke CEK_LCD
CLR ENABLE      ; setelah Port 2.7 '0' ENABLE = '0'
RET             ; return
```

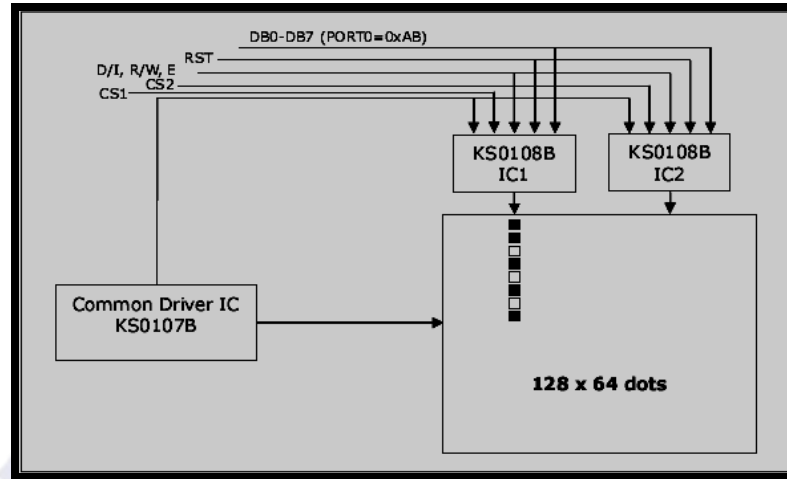
**Subrutin reset modul LCD:**

RESET:

```
CLR ENABLE       ; ENABLE = '0'
CLR RES          ; RES = '0'
SETB RES         ; RES = '1'
RET              ; return
```

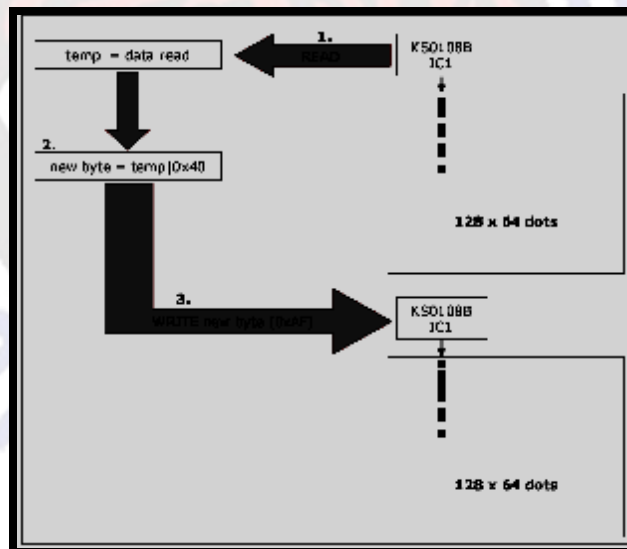
Penjelasan berikutnya adalah algoritma perangkat lunak untuk mengakses LCD, menghapus layar LCD, mengkonversi dan menggambar data hasil pengukuran. Rutin hapus layar diperlukan karena tiap titik pada layar LCD adalah alamat memori penyangga internal modul LCD yang bersifat tetap jika catu daya tidak dimatikan, sehingga pada waktu menampilkan data hasil pengukuran yang lebih baru, data lama perlu dihapus dari layar dan tidak dapat secara otomatis dilakukan oleh modul tetapi hanya dapat diprogram dari mikrokontroler.

Pada gambar 7 ditunjukkan jika pada layar pertama dari modul, pada salah satu halaman di salah satu kolom ditulis data AB heksadesimal atau 10101011 biner.



Gambar 7. Contoh Pemberian Data 10101011biner

Pada gambar 8 diberikan contoh dari gambar 7 jika data pertama akan di 'OR' kan dengan data yang lebih baru yaitu 04 heksadesimal.



Gambar 8. Proses meng'OR'kan data

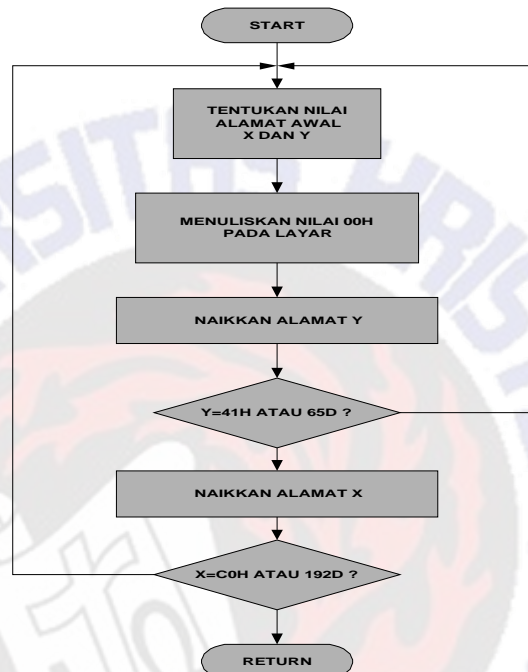
Untuk membersihkan layar LCD maka dapat dibuat Algoritma menghapus layar LCD adalah sebagai berikut.

1. Tentukan nilai alamat awal X dan Y. Penulis mulai dari X=0 dan Y=0.
2. Siapkan data 8 bit berisi 0 desimal.
3. Panggil prosedur menulis LCD dengan data tulis 8 bit yang telah disiapkan.
4. Alamat X=0 dan Y=0 telah berisi 0 desimal atau kosong.
5. Selanjutnya menuliskan 0 desimal pada X=0 dan Y=1 dengan data yang sama, cara menuliskan hanya dengan memanggil prosedur menulis LCD tidak perlu mengirimkan data alamat yang diinginkan karena setiap LCD melakukan instruksi penulisan atau pembacaan, alamat Y akan dinaikkan satu secara otomatis oleh perangkat keras modul LCD.



6. Langkah 5 dilakukan terus menerus sebanyak 64 kali sehingga dalam 1 halaman X=0 datanya telah berisi 0 semua.
7. Untuk selanjutnya diulangi langkah 4 dengan alamat awal X=1 dan Y=0.
8. Kemudian mengulangi langkah 5 dan 6 sampai 1 layar penuh X=0 hingga X=7 dan kolom Y=0 hingga Y=63 berisi 0 desimal.
9. Untuk layar 2 dilakukan sama seperti pada langkah 1-8 dengan mengatur nilai CS1='1' dan CS2='0'. Kemudian kembali ke program utama.

Diagram alir untuk menghapus layar LCD ditunjukkan pada gambar 9 berikut ini.



Gambar 9. Diagram Alir Rutin Menghapus Layar LCD

### Subrutin menghapus layar LCD:

CLRSCR:

```

CLR DI          ; DI = '0' à instruksi
CLR RW          ; RW = '0' à tulis
MOV R0,#10111000B
MOV DATA_LCD,R0 ; set ke halaman 0
SETB ENABLE    ; ENABLE = '1'
CLR ENABLE     ; ENABLE = '0'
CALL CEK_LCD   ; lompat ke subrutin cek_lcd, apakah sibuk?
    
```

POSISIY:

```

CLR DI          ; DI = '0' à instruksi
CLR RW          ; RW = '0' à tulis
MOV R1,#01000000B ; set ke kolom 0
MOV DATA_LCD,R1
SETB ENABLE    ; ENABLE = '1'
CLR ENABLE     ; ENABLE = '0'
CALL CEK_LCD   ; lompat ke subrutin cek_lcd, apakah sibuk?
MOV R1,#0
    
```

CLR1:

```

MOV DATA_LCD,#00000000B ; siapkan data 00H
SETB DI                 ; DI = '1' à data
CLR RW                  ; RW = '0' à tulis
SETB ENABLE
CLR ENABLE
    
```

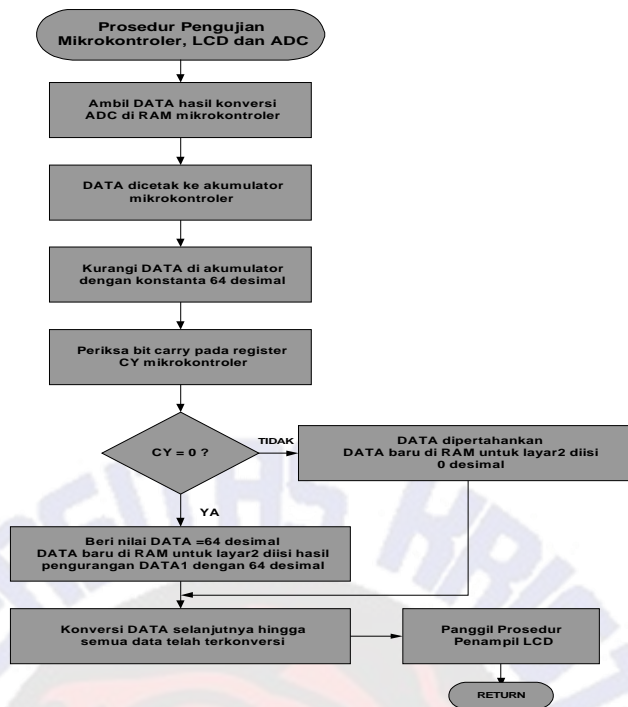
```

CALL CEK_LCD           ; lompat ke subrutin cek_lcd, apakah sibuk?
INC R1                 ; R1 = R1+1 naikkan posisi kolom
CJNE R1,#01000001B,CLR1 ; bandingkan jika R1 = 41H kolom = 64
INC R0                 ; R0 = R0+1 naikkan posisi halaman
MOV DATA_LCD,R0
CLR DI                 ; DI = '0' à instruksi
CLR RW                 ; RW = '0' à tulis
SETB ENABLE
CLR ENABLE
CALL CEK_LCD           ; lompat ke subrutin cek_lcd, apakah sibuk?
CJNE R0,#0C0H,POSISIY ; bandingkan jika X=7
RET
    
```

Perangkat lunak untuk pengujian yang dibuat mempunyai algoritma sebagai berikut.

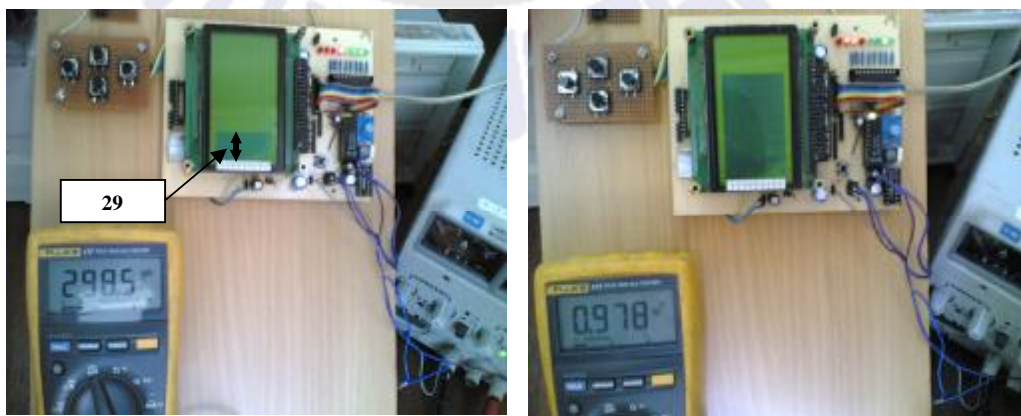
1. LCD Grafik secara horisontal dibagi menjadi 8 halaman. Masing –masing halaman terdiri dari 64 kolom. Tiap kolom dari tiap halaman memiliki 8 titik yang hanya bisa diakses dalam 8 titik sekaligus.
2. Port 3 disiapkan untuk mengambil data digital dari ADC.
3. Pengambilan data dari ADC dilakukan sebanyak 8 kali untuk 8 halaman pada layar LCD disimpan dalam RAM mikrokontroler.
4. Karena satu layar penuh LCD dibagi menjadi dua dan hanya bisa diakses satu persatu maka maksimum titik pada layar hanya 64.
5. Jika data dari ADC bernilai 64 atau lebih maka layar pertama akan menunjukkan nilai maksimum dan sisanya tak dapat ditampilkan. Untuk menampilkan sisa dipakai layar kedua. Perangkat lunak akan mengkonversi 8 buah data masing-masing sebesar 8 *byte*. Tiap data kan dibandingkan dengan 64 desimal, jika data lebih kecil atau sama dengan 64, nilai variabel data akan dipertahankan dan nilai variabel untuk layar kedua diisi nol. Jika data lebih besar dari 64 maka data untuk layar pertama diisi 64 desimal dan data untuk layar kedua diisi hasil pengurangan data awal dengan 64 desimal.
6. Setelah proses konversi selesai maka data untuk layar pertama dan kedua di RAM mikrokontroler menjadi 16 *byte*.
7. Perangkat lunak akan mulai menampilkan pada layar pertama dan kemudian layar kedua. Tiap kenaikan 1 titik pada layar LCD menunjukkan kenaikan tegangan masukan sebesar 10 milivoltDC.

Diagram alir dari pengujian mikrokontroler, LCD Grafik dan ADC ditunjukkan pada gambar 10.

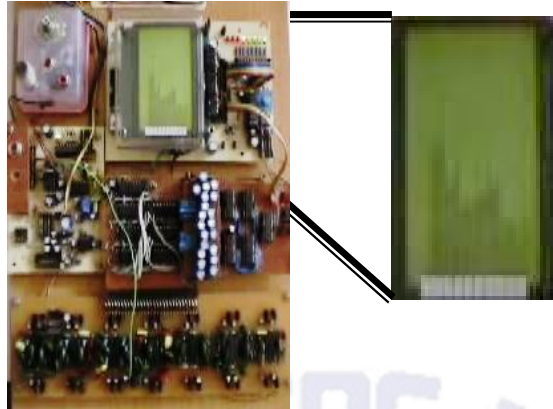


Gambar 10. Diagram Alir Pengujian Modul Mikrokontroler, LCD dan ADC

Pengujian modul mikrokontroler, modul LCD Grafik dan ADC 8 bit dilaksanakan dengan memberi masukan ke pin masukan analog dari ADC, maka diharapkan pada layar LCD Grafik akan ditampilkan nilai tegangan masukan yang diwakili 1 titik untuk tiap 10 milivoltDC. Sebagai contoh ditunjukkan pada gambar 11 yaitu tampilan LCD saat ADC diberi masukan tegangan 298,5 milivoltDC, jumlah titik vertikal yang tertampil di LCD adalah 29 titik. Sedangkan saat diberi tegangan 978 milivoltDC dibagi 10 milivoltDC yaitu 97,8 titik oleh ADC dikonversi menjadi 98 titik vertikal atau 64 titik pada layar pertama dan sisanya 34 titik pada layar kedua



Gambar 11. Tampilan LCD Grafik Ketika Tegangan Masukan ADC 298,5 dan 978 milivoltDC



Gambar 12. Aplikasi LCD Grafik untuk Penganalisa Spektrum Audio dan Akustik 1/3 oktaf

### Daftar Pustaka

- [1] Mackenzie, I. Scott, *The 8051 Microcontroller*, 2nd Edition. Prentice Hall. 1995.
- [2] Leung, John, *Chapter 8. Interfacing a JHD12864J Graphic Module to AT89S52*, [www.TechToys.com.hk](http://www.TechToys.com.hk), 2006
- [3] *Design of a Graphical LCD Driver and Educational LCD Primer, A Design Project Report Presented to the Engineering Division of the Graduate School of Cornell University in Partial Fullfilment of the Requirement for the Degree of Master of Engineering (Electrical)*, by Lucas L. Delaney, Project Advisor Dr. Bruce R. Land, Degree Date May 2004.
- [4] Datasheet ADC0820, 8-Bit Half Flash ADC, National Semiconductor.

### Riwayat Penulis

#### Deddy Susilo

Lahir di Semarang pada tanggal 8 September 1981, lulus dari Fakultas Teknik Program Studi Teknik Elektro, Universitas Kristen Satya Wacana, Salatiga pada bulan Agustus 2006. Mulai 1 September 2006 bekerja sebagai dosen di Fakultas Teknik Program Studi Teknik Elektro, Universitas Kristen Satya Wacana, Salatiga